IN THE CLAIMS:

Please cancel claims 16 and 19, and amend the claims as follows:

Claim 1 (Currently Amended): Method for scheduling the service of a thread, said method comprising the steps of:

masking interrupts from one or more hardware devices in order to ignore interrupts for other threads;

acquiring a latency information associated with the thread, wherein the latency information indicates a time at which the thread needs to be processed;

unmasking interrupts from the one or more hardware devices in order to detect interrupts for the other threads; and

rearranging an order in which the thread and the other threads will be serviced in a single queue to schedule the thread for processing in accordance with said latency information.

Claim 2 (Previously Presented): The method of claim 1, wherein said latency information is computed based on a buffer size or display rate.

Claim 3 (Currently Amended): The method of claim [[2]]1, ~~wherein said~~ further comprising computing the time at which the thread needs to be processed by summing the latency information ~~is representative of time units~~ with a current time.

Claim 4 (Currently Amended): The method of claim [[3]]1, wherein said latency information represents a time duration that is necessary to service the thread.

Claim 5 (Currently Amended): The method of claim [[3]]1, wherein said latency information represents a maximum time ~~at which said scheduled thread will be processed~~ allowed before a first buffer will be emptied and a read operation will switch to process a second buffer.

Claim 6 (Currently Amended): The method of claim [[3]]1, wherein said latency information represents a time duration that is necessary to setup the thread to perform interrupt processing for the thread.

Claim 7 (Currently Amended): The method of claim [[3]]1, wherein said latency information is dependent on a hardware constraint for one of the one or more hardware devices.

Claim 8 (Currently Amended): The method of claim [[3]]1, wherein said latency information is provided by a device driver.

Claim 9 (Currently Amended): Apparatus for scheduling the service of a thread, said apparatus comprising:

means for receiving an interrupt from a hardware device;

means for masking interrupts from one or more hardware devices in order to ignore interrupts for other threads;

means for acquiring latency information associated with the interrupt, wherein the latency information indicates a time at which the thread needs to be processed;

means for unmasking interrupts from the one or more hardware devices in order to detect interrupts for the other threads; and

means for rearranging an order in which the thread and the other threads will be serviced in a single queue to schedule the thread to process the interrupt in accordance with said latency information.

Claim 10 (Currently Amended): The apparatus of claim 9, ~~wherein said~~ further comprising computing the time at which the thread needs to be processed by summing the latency information ~~is representative of time units~~ with a current time

Claim 11 (Previously Presented): The apparatus of claim 9, wherein said latency information is dependent on a hardware constraint for one of the one or more hardware devices.

Claim 12 (Original): The apparatus of claim 11, wherein said hardware constraint is a size of a buffer.

Claim 13 (Original): The apparatus of claim 11, wherein said hardware constraint is a fullness of a buffer.

Claim 14 (Previously Presented): The apparatus of claim 11, wherein said hardware constraint is dynamically computed based on a buffer size or display rate.

Claim 15 (Original): The apparatus of claim 9, wherein said latency information is generated by a device driver associated with the hardware device.

Claim 16 (Canceled)

Claim 17 (Currently Amended): The method of claim 1, further comprising toggling an interrupt line.

Claim 18 (Previously Presented): The method of claim 1, further comprising:
determining the thread should be activated; and
activating the thread for processing.

Claim 19 (Canceled)

Claim 20 (Previously Presented): The method of claim 1, further comprising determining ~~wherein said latency information represents a time duration that is used to determine~~ when the thread should be activated for processing using the latency information or an absolute time that is computed using the latency information.

Claim 21 (Currently Amended): The method of claim 1, further comprising:

creating the thread ~~for interrupt processing~~ prior to the steps of masking, acquiring, unmasking, and rearranging when one of the one or more hardware devices is initialized, wherein the thread is created for use during processing of a first interrupt that the one of the one or more hardware devices is configured to generate; and

freeing the thread when the one of the one or more hardware devices is shut down.

Claim 22 (Currently Amended): The method of claim 21, further comprising:

creating an additional thread ~~for interrupt processing of another interrupt that the one of the one or more hardware devices is configured to generate~~, wherein a first interrupt identification number is associated with the thread and a second interrupt identification number that is different than the first interrupt identification number is associated with the additional thread and the additional thread is created for use during processing of a second interrupt that the one of the one or more hardware devices is configured to generate; and

freeing the additional thread when the one of the one or more hardware devices is shut down.

REMARKS

This amendment is submitted as a full and complete response to the Office Action dated April 5, 2007. Reconsideration and allowance of the claims is requested. The independent claims have been amended in each instance to more distinctly claim the subject matter that the Applicants regard as the invention. An explanation of each amendment appears below.

Claims 1-15, 17-18, and 20-22

Claims 1-15, 17-18, and 20-22 are rejected under 35 U.S.C. § 112 as being indefinite for failing to particularly point out and distinctly claim the subject matter which the Applicant regards as the invention. Limitations reciting "the thread" are considered indefinite. Claim 1 recites "a thread" and "other threads," which are subsequently referenced as "the thread" and "the other threads," respectively. Applicant respectfully disagrees with the Examiner's assertion that the references to "the thread" and "the other threads" are unclear. In particular, consistently throughout all of the claims, "the thread" clearly refers to "a thread" and "the other threads" clearly refers to "other threads."

Applicant is amending claims 5, 21, and 22 to comply with the written description requirement of 35 U.S.C. § 112. In particular, in claim 5 "said scheduled thread" has been changed to "the thread," in claim 21 "the thread for interrupt processing" has been changed to "the thread," and in claim 22 "an additional thread for interrupt processing" has been changed to "an additional thread." Furthermore, "an interrupt" and "another interrupt" has been changed to "a first interrupt" and "a second interrupt," respectively, in claims 21 and 22.

Claims 3, 5, 10, and 20 are rejected under 35 U.S.C. § 112 for failing to further limit the parent claim. Claims 3 and 10 are amended to include the limitation of summing the latency information with a current time to compute an absolute time, as described in paragraphs [0013] and [0024] of the present application. Claim 5 is amended to clarify the limitation of the latency information as a maximum time that is allowed before a first buffer will be emptied and a read operation will switch to process a second buffer, as described in paragraph [0044] and Figure 3 of the present application.

Claim 20 is amended to clarify the limitation of determining a time at which the thread is activated based on the latency information, as described in paragraph [0045] and Figure 4 of the present application.

The Examiner has objected to the drawings for failing to show every feature specified in claims 21 and 22. The features of claims 21 and 22 are described in paragraphs [0031] and [0032] of the application, respectively. As described, the claimed features are shown as NvArch CODE 264 in Figure 2 of the application as filed.

Claims 1, 3-6, 9-10, and 20

Claims 1, 3-6, 9-10, and 20 are rejected under 35 U.S.C. 102(b) as being anticipated by *Zolnowsky* (U.S. Patent No. 5,826,081). These rejections are respectfully traversed.

Applicant is amending independent claims 1 and 9 to include the limitation of a single queue that represents the order in which the thread and the other threads will be serviced. As described in paragraph [0044] of the present application, the scheduler "rearranges various threads in the queue such that threads that require faster service based on latency will be processed ahead of other threads that can wait a bit longer." Since the threads are scheduled based on the hardware-provided latency information, low priority processes, e.g., non-interrupt based scheduled threads, and high priority threads (e.g., interrupt threads) can be arranged in a single queue. As stated in the paragraph [0004] of the background section of the present application, some prior art systems use two queues, where one queue is dedicated to high priority processes and another queue is dedicated to address low priority processes in order to meet real-time processing constraints.

Zolnowsky teaches using several queues to schedule operations. In particular, each processor has a queue and a global dispatch queue is used for higher priority real time threads, as described in column 6, lines 27-33. Nowhere does *Zolnowsky* teach or suggest rearranging the order in which threads will be serviced from a single queue. For these reasons, Applicant submits that amended claims 1 and 9 are in condition for allowance and respectfully request withdrawal of the 35 U.S.C. § 102(b) rejection of these claims. Since claims 2-8, 17-18, and 20-22 depend from allowable claim 1 and

claims 10-15 depend from allowable claim 9, these claims also are in condition for allowance.

Claims 1-15, 17-18, and 20

Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Ramakrishnan* (U.S. Patent No. 6,085,215) and "Process Management – Scheduling" ECEN5043 Software Engineering of Multi-Program Systems, University of Colorado, Boulder ("ECEN5043"). These rejections are respectfully traversed.

Applicant is canceling claims 16 and 19. Amended claims 1 and 9 include the limitation of rearranging the order in which the thread and the other threads will be serviced to schedule the thread for processing in accordance with latency information. Importantly, claims 1 and 9 are also amended to clarify that the latency information indicates a time at which the thread needs to be processed. All of the threads are then rearranged to accommodate the time at which the thread needs to be processed.

The Examiner relies on *ECEN5043* for the teaching of rearranging the order in which the thread and the other threads will be serviced to accommodate latency requirements. However, *ECEN5043* teaches scheduling based on a "shortest job" rather than a time at which a particular thread needs to be processed. Specifically, *ECEN5043* describes estimating how long a process will take and selecting the shortest job first when the jobs have equal priority. Scheduling threads based on job length is very different than scheduling the threads based on when a thread needs to be processed, as recited in amended claims 1 and 9 of the present application. This failure precludes the combination of *Ramakrishnan* and *ECEN5043* from rendering obvious amended claims 1 and 9. For these reasons, Applicant submits that amended claims 1 and 9 are in condition for allowance and respectfully request withdrawal of the 35 U.S.C. § 103(a) rejection of these claims. Since claims 2-8, 17-18, and 20-22 depend from allowable claim 1 and claims 10-15 depend from allowable claim 9, these claims also are in condition for allowance.